

Temperature-Resilient Analog Neuromorphic Chip in Single-Polysilicon CMOS Technology

Tommaso Rizzo , Member, IEEE, Sebastiano Strangio , Senior Member, IEEE, Alessandro Catania , Senior Member, IEEE, and Giuseppe Iannaccone , Fellow, IEEE

Abstract—In analog neuromorphic chips, designers can embed computing primitives in the intrinsic physical properties of devices and circuits, heavily reducing device count and energy consumption, and enabling high parallelism, because all devices are computing simultaneously. Neural network parameters can be stored in local analog non-volatile memories (NVMs), saving the energy required to move data between memory and logic. However, the main drawback of analog sub-threshold electronic circuits is their dramatic temperature sensitivity. In this paper, we demonstrate that a temperature compensation mechanism can be devised to solve this problem. We have designed and fabricated a chip implementing a two-layer analog neural network trained to classify low-resolution images of handwritten digits with a low-cost single-poly complementary metal-oxide-semiconductor (CMOS) process, using unconventional analog NVMs for weight storage. We demonstrate a temperature-resilient analog neuromorphic chip for image recognition operating between 10°C and 60°C without loss of classification accuracy, within 2% of the corresponding software-based neural network in the whole temperature range.

Index Terms—Analog computing, analog neural networks, analog non-volatile memory, computing in-memory, vector-matrix multiplier, neuromorphic engineering.

I. INTRODUCTION

REAL-TIME critical applications require low-latency inference, which can be obtained if artificial neural networks (ANNs) are directly embedded in sensors and in edge devices, with limited compute and energy resources. Inference is computationally demanding and inefficient when performed with classical von Neumann architectures, where processing is separated from memory [1], [2]. Analog neuromorphic chips based on analog in-memory computing (AIMC) architectures have the potential to overcome the bottlenecks of traditional approaches [3]–[5], through the maximally parallel one-shot operation of vector-matrix multipliers (VMMs): designers are able to intrinsically embody computing primitives in device physics and in circuit topology, and hence obtain at the

This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Tommaso Rizzo is with Quantavis s.r.l., Largo Padre Renzo Spadoni, 56126 Pisa, Italy and with the Dipartimento di Ingegneria dell'Informazione (DII), Università di Pisa, 56122 Pisa, Italy; Sebastiano Strangio is with the Dipartimento di Ingegneria dell'Informazione (DII), Università di Pisa, 56122 Pisa, Italy; Alessandro Catania is with the Dipartimento di Ingegneria dell'Informazione (DII), Università di Pisa, 56122 Pisa, Italy; Giuseppe Iannaccone is with the Dipartimento di Ingegneria dell'Informazione (DII), Università di Pisa and with Quantavis s.r.l., Largo Padre Renzo Spadoni, 56126 Pisa, Italy (corresponding author: (e-mail: giuseppe.iannaccone@unipi.it)

This work was partially supported by the EC Horizon 2020 Research and Innovation Programme under GA QUEFORMAL 829035 and under GA AUTOCAPSULE 952118, and by the Italian MUR under the Forelab project of the “Dipartimenti di Eccellenza” programme.

same time elegant, compact, and massively parallel computing elements [6]–[8].

Analog neuromorphic chips use transistors and locally instantiated non-volatile memories operated in sub-threshold conditions [9] to reach ultra-low power consumption and a large dynamic range [10]–[14]. They can exploit the fact that ANNs can achieve high inference accuracy even under reduced precision [15] to effectively address the susceptibility of analog computing to noise and non-linearity. This enables analog neuromorphic chips to achieve energy efficiency improvements of up to two orders of magnitude compared to digital accelerators [15]–[18].

Several NVM technologies have recently been considered for AIMC implementations [8], [19]. Some of them are based on emerging materials: magnetic tunnel junctions [20], [21], ferroelectric devices [22]–[25], and synaptic memory transistors based on transition-metal dichalcogenides (e.g. MoS₂ [26]–[29]). Given that device yield is still limited and integration with CMOS is challenging for some materials, in these cases the full-network operation is implemented only through software-level or mixed-hardware approaches. On the other hand, multicore accelerators integrating in the same chip both the memristor-based NVM weights (resistive random access memory, RRAM [30]–[33] or phase change memory, PCM [34], [35]) along with the other processing circuits, have been realized taking advantage of CMOS compatibility. However, since these types of memory are inherently bistable, multiple cells are used to store each individual weight to achieve sufficient analog resolution, such as two RRAM cells [31] or four PCM cells [35]. A sophisticated programming scheme has recently been proposed to improve the programming resolution and reproducibility of RRAMs [36], but memory retention is limited by conductance relaxation [30].

Well-established CMOS NVM technologies, consisting of commercial NVM arrays of double-poly NOR flash [10] or single-poly Y-Flash [37], have also been proposed for AIMC applications, and are at the basis of larger scale chips by startups in the field [38]. The nonlinear current-voltage behavior of such memristive cells is a known challenge: as an example, only binary inputs are accepted by the 12×8 AIMC Y-Flash crossbar array [11], [13] to avoid introducing distortion in the multiplication operations.

As is very typical of subthreshold analog circuits, sensitivity of AIMC circuits to temperature is a big challenge: their behavior as a function of temperature is normally reported only for some memory retention tests, while no experimental data have been shown on inference accuracy degradation of ANNs with temperature variation. As for AIMC based on

CMOS flash, only few temperature compensation approaches have been investigated through circuit simulations or partial multiplier implementations [39]–[42], and similar techniques have been proposed for mixed-signal in memory computing based on SRAM digital weights [43]–[45].

We have devised a technique to adapt the driving signals of the neuromorphic chip in response to temperature variation in order to compensate for the temperature sensitivity of the analog non volatile memories. This enables us to demonstrate a temperature-resilient analog CMOS multi-layer ANN, with multi-bit equivalent precision for inputs, weights and outputs of each layer of neurons.

The ANN analog chip, described in Section II, is fabricated with a single-poly 180 nm CMOS process, embedding two 16×16 analog time-domain VMMs (TD-VMMs), with the analog weights stored in two-terminal single-transistor floating-gate (1T-FG) memory cells [14]. The TD-VMM consists of a 16×16 weight crossbar array which is realized with 256 densely placed 1T-FG cells, each obtained using 3.3 V n-type minimum-size transistors with a 7 nm-thick SiO_2 gate oxide and a single-poly gate that is left floating (only $1.72 \mu\text{m}^2$ per each 1T-FG cell). We use the 1T-FG cells as programmable current sources: the associated weight is the current flowing through the 1T-FG cell when it is active, i.e., when a con-

stant voltage amplitude V_{READ} is applied between drain and source. In this way the nonlinear current-voltage characteristic of 1T-FG cells does not affect multiplication accuracy, and therefore we experimentally achieve an equivalent number of bits (ENOB) of up to 4.7 bits for the TD-VMM weights and up to 5.7 bits for the TD-VMM outputs.

The temperature compensation method for the VMM and for the weights stored in the analog non-volatile memory array is described in Section III. We analyze the intrinsic temperature dependence of the weight stored in the memory array and we devise an adjustment technique for the driving voltages of the VMM that makes the weights almost independent of temperature in a broad temperature range.

As a proof of concept, and to test the inference accuracy on a known benchmark, we have trained the two-layer ANN for the classification of a low-resolution (4×4 pixels) version of the handwritten digit MNIST database [46], reaching a classification accuracy of 83.1% at the temperature of 30°C (results are shown in Section IV). We have rigorously characterized the chip to assess its temperature sensitivity and the time retention of the analog NVM. We verify that a classification accuracy higher than 81.5% can be achieved in a $10 \div 60^\circ\text{C}$ temperature range (81.56 % at 10°C, 83.03% at 60°C), thanks to the adaptive compensation of V_{READ} in response to temperature

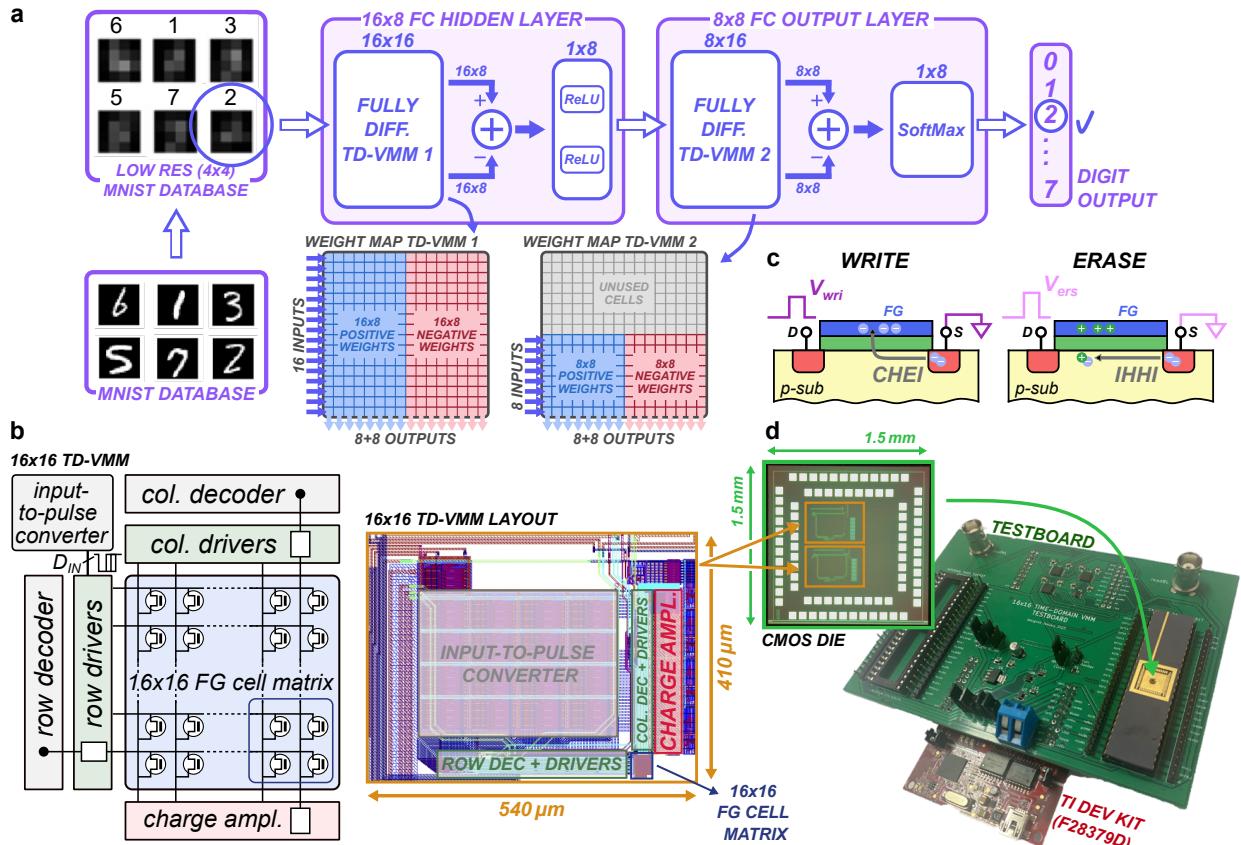


Fig. 1. Multi-layer fabricated neural network. (a) Architecture of the multi-layer neural network performing image classification on a low-resolution (4×4) version of MNIST database, with detail of the mapped weights on the two TD-VMMs used in the hidden and output fully-connected layers; (b) architecture and layout of the designed analog 16×16 time-domain VMM, with details of row and column decoders and drivers, input-to-pulse converter, charge amplifiers and NVM FG cell matrix; (c) sketch of the physical phenomena involved when writing (hot electron injection) or erasing (impact-ionisation hot hole injection) a 1T-FG cell; (d) the fabricated chip was packaged and tested using a custom PCB and commercial development kit.

variation, and for 7 days at 20°C, or for more time if a memory refresh is performed on a weekly basis.

II. ANALOG NEURAL NETWORK CHIP

A. Analog neural network architecture

The architecture and the concept of operation of the analog neural network are shown in Fig. 1a: the inputs are low-resolution images of handwritten digits from ‘1’ to ‘8’ (4×4 pixels, 32-tone grayscale representation), which are a reduced version of the MNIST dataset images. The input image is passed as an 80-bit [(4×4) pixels \times 5 bits] digital stream to the analog chip, where it is converted into 16 voltage pulses (one per pixel) of fixed amplitude V_{READ} and width proportional to the gray level, which represent the input vector of the first fully-connected 16×8 hidden neuron layer, consisting of an analog TD-VMM followed by a Rectified Linear Unit activation function acting on the 8 sampled outputs; after that, the resulting data are processed by a fully-connected 8×8 output neuron layer, again consisting of a second analog TD-VMM followed by a softmax activation function that performs the final 8-digit classification.

To address the need of both positive and negative weights, Fig. 1a illustrates the mapping schemes adopted for both hidden- and output-layer weights. To map the positive and negative weights of the 16×8 reference (target) hidden layer into the corresponding hardware TD-VMM, we exploit the full 16×16 matrix of the first TD-VMM to write positive weights on its left half (with a value of ‘0’ in correspondence to the location of negative weights) and the absolute value

of negative weights on its right half (with a value of ‘0’ in correspondence to positive weights). Therefore, we are able to perform the 16×8 operations – with signed number representations – by subtracting the 8 right-side outputs from the left-side ones. Similarly, the reference output-layer positive and negative weights are mapped into an 8×16 matrix within the second hardware TD-VMM, to address the related signed-representation 8×8 VMM operation. For the sake of clarity, from now on we distinguish between unsigned (1T-W) and signed (2T-W) analog weights, as obtained by the difference between two unsigned 1T-W cells implementing the same weight.

Fig. 1b focuses on the top-level architecture and layout of the analog TD-VMM: its core is the 16×16 memory matrix realized with single-poly 1T-FG cells, together with an array of 16 charge amplifiers – one for each column – where the compute-in-memory operation takes place [15]. In addition, an array of digital input-to-pulse converters is used to feed the input pulse data, and row/column decoders and drivers enable the implementation and selection of the proper operation mode (comprising also the programming and reading of the NVM matrix). Each cell can be selected via row and column decoders, whereas row and column drivers are exploited to properly bias selected and non-selected cells.

The analog weights are programmed (Fig. 1c) using a series of voltage pulses of different amplitudes on the drain, with the source grounded, using a program and verify scheme [14]. Some more details are discussed in Appendix A. Fig. 1d shows the experimental chip that includes the two TD-VMMs,

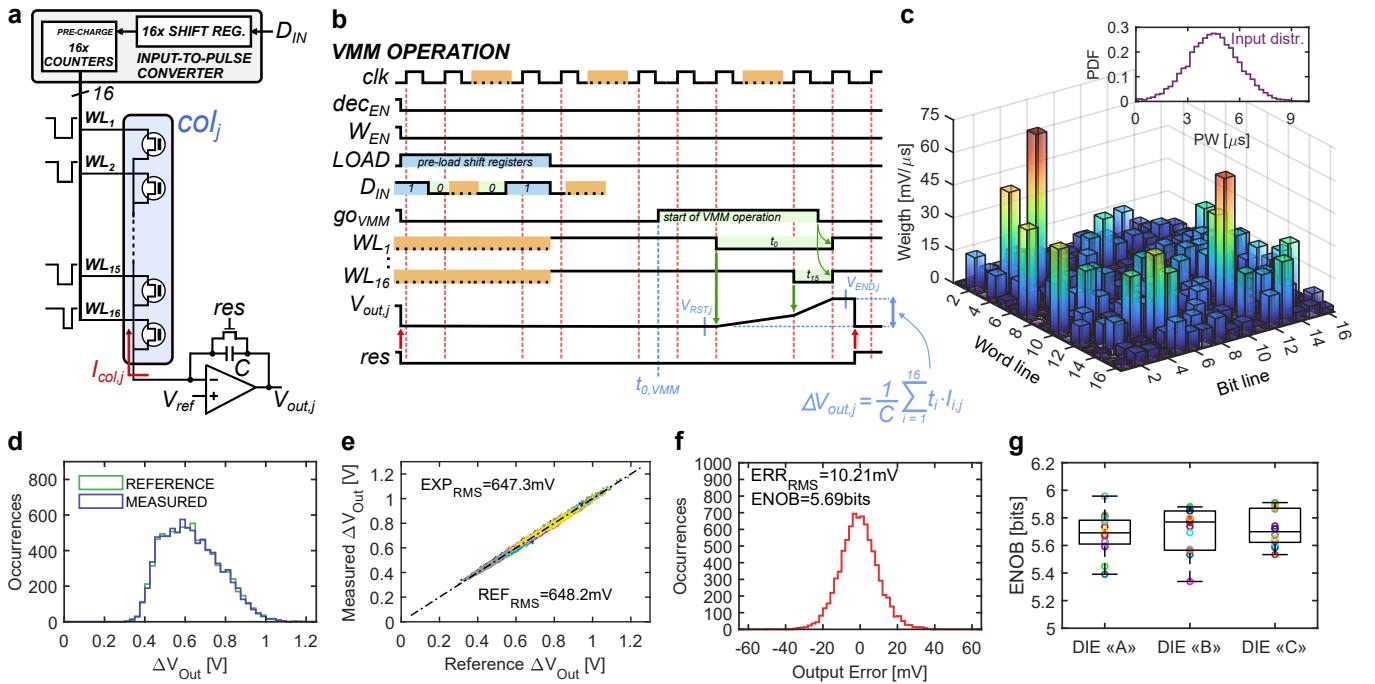


Fig. 2. TD-VMM operation and precision characterisation: (a) Detail of the MAC operations performed by a column (BL) of the TD-VMM; (b) timing diagram of the VMM operation; (c) spatial map of the native weights of a typical 16×16 crossbar array (random pulse-width input vector distribution shown in the inset); (d) histograms of the measured and theoretical VMM outputs for a given set of random input data, with (e) corresponding scatter plot of measured vs. theoretical outputs (ERR = $\Delta V_{\text{OUT},\text{EXP}} - \Delta V_{\text{OUT},\text{REF}}$); (f) and extracted output errors (ERR = $\Delta V_{\text{OUT},\text{EXP}} - \Delta V_{\text{OUT},\text{REF}}$); (g) ENOB extracted for each BL column of three different dies.

with dual bonding option for debug purposes. In addition, a test setup, composed of a custom printed circuit board with die socket and of a commercial evaluation board (TI LAUNCHXL-F28379D), was implemented to run the characterization and testing of the multi-layer ANN chip.

We limit our analysis to the proof-of-concept network of Fig. 1a due to the available hardware, as our demonstrator chip contains two 16×16 VMMs. Let us stress that the proposed temperature compensation mechanism is based on a general concept that can be effective also on larger networks. Future works will address the design and fabrication of larger networks, enabling classifiers for high-resolution datasets and addressing practical scaling up issues and half-selection issues (discussed more in detail in Appendix A).

B. Time-Domain Vector-Matrix Multiplier

To illustrate the operation of the proposed TD-VMM, Fig. 2a shows the details of a VMM column implementation. The inputs are provided as voltage pulses at each row – word-line (WL) – connected to the 1T-FG cell source terminals. These pulses have an amplitude that ensures a constant drain-to-source voltage V_{READ} is applied to the corresponding 1T-FG cell, and a pulse width corresponding to the analog input data. As a result, the drain currents of the 1T-FG cells have a rectangular-shape waveform: for each transistor cell, the on-current level is the analog weight, whereas the integral of the corresponding current over time (i.e., the total charge) represents the physical product between the pulse width (proportional to the pixel gray level) and the on-current (proportional to the weight). The drain terminals of all 1T-FG cells in the same column are connected through the column driver to the corresponding bit-line (BL) and to the input of the related charge amplifier: the sum of the charges flowing through the cells in the same column j (i.e. the sum of the products of analog inputs by analog weights) is conveyed at the inverting input of the charge amplifier, and is converted into

an output voltage ($V_{\text{out},j}$). With reference to Fig. 2a, showing the details of a VMM column implementation during VMM operation, the decoders are deactivated, since all the 16×16 matrix cells operate in parallel; the drivers make sure that each row – word-line (WL), connected to the 1T-FG cell source terminals – sees its relative input time pulse, which activates with a fixed drain-to-source voltage the corresponding row of 1T-FG cells for the time of its duration. The timing diagram of the VMM operation is depicted in Fig. 2b and described in detail in Appendix B.

C. Temperature Characterization Methodology

Fig. 3 reports a flowchart highlighting tools and techniques used in the present work: we have trained a two layer neural network in Matlab and the trained weights have been programmed into the chip hidden- and output-layer at nominal temperature $T_0 = 30^\circ\text{C}$ and $V_{\text{READ}} = 1.15$ V. As the weights vary exponentially when changing the temperature, the temperature compensation mechanism exploits the variation of the V_{READ} to adjust the weight values. Therefore, we have evaluated the experimental weights programmed in the previous step by varying the temperature in the $10 \div 60^\circ\text{C}$ range and the V_{READ} in the $1.06 \div 1.21$ V range. In this way, by plotting extracted weights (W_{EXP}) versus target ones (W_{TGT}), we have been able to quantify the magnification M_T (first order effect) and the ENOB, using the definition inherited from the analog-digital converter characterization protocol [47]: $(\text{SNDR}_{\text{dB}} - 1.76)/6.02$, where SNDR (signal to noise and distortion ratio) is here calculated as the ratio of the W_{TGT} RMS value to the $\Delta W = W_{\text{EXP}}/M_T - W_{\text{TGT}}$ RMS error, which is caused by both noise and nonlinearity. For the same temperature and V_{READ} conditions, the implemented neural network has been fully characterized by processing all the low-resolution MNIST images of the test dataset, and the output results have been extracted and compared with the correct results.

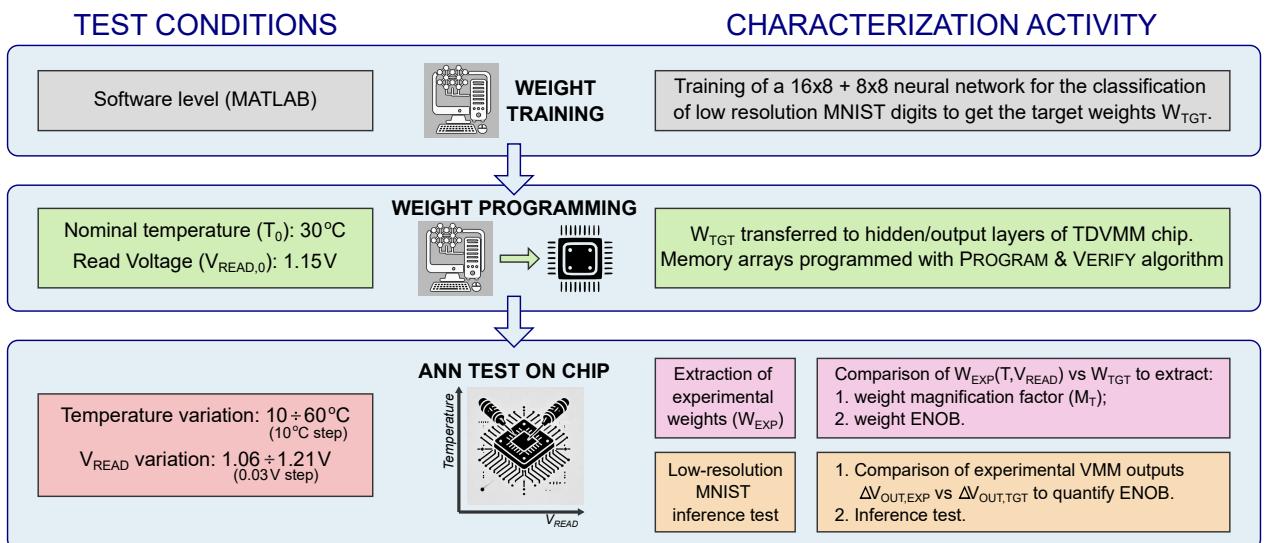


Fig. 3. Flowchart of the tools and techniques used in the present work.

III. TEMPERATURE COMPENSATION OF ANALOG WEIGHTS AND OF VMM OPERATION

Due to the analog nature of the neuromorphic chip, CMOS temperature dependent parameters could undermine the inference accuracy of the neural network, when the actual temperature T deviates from the nominal temperature T_0 , i.e. the temperature of the chip when the analog weights have been programmed with the program and verify scheme.

Although FG memory cells operating in the sub-threshold regime are highly sensitive to temperature, only few papers investigate compensation approaches, typically through circuit simulations or partial implementations: in [39], a translinear loop-based FG current reference circuit was proposed, achieving linear FG charge-to-current proportionality while reducing temperature sensitivity; however, each cell is large (consisting of one FG cell, four transistors, and one resistor) and cannot be considered a viable option for large density VMMs. Refs. [40] and [41] investigate current mirror FG cell topologies, with weights embedded in the current magnification factors with respect to the input current of a reference FG cell. Since the control gate voltage is not fixed but it is generated by the input cell, it is dynamically adjusted in reaction to temperature variations. However, besides the special case of unitary current gain, significant sensitivity remains for large or small weights. Ref. [42] presents an oversimplified analysis of an AIMC computing chip, using simulations on standard 180 nm CMOS transistors to emulate FG cells; the proposed VMM is implemented with transistors operating in the linear region, mapping weights onto the threshold voltage difference of two coupled transistors (it is not clear how the threshold voltage V_{th} is programmed without an FG) and inputs to the drain-source voltage. By making the difference of the output currents to get the output result, they mitigate V_{th} variations, leaving only mobility dependence on temperature. Other temperature compensation strategies have been proposed for mixed-signal in memory computing based on SRAM digital weights [43]–[45]. Regarding alternative weight memory representation methods, different memory cell technologies exhibit distinct temperature-dependent behaviors due to the varying physical mechanisms governing conduction, therefore require ad hoc compensation techniques.

Fig. 4 analyzes how accurately the target weights W_{TGT} are mapped into the two TD-VMM memory cores as experimental analog weights W_{EXP} , with emphasis on temperature behavior; a voltage amplitude adaptive compensation is then proposed to improve weight accuracy.

At nominal temperature T_0 , analog weight programming grants good accuracy, as confirmed by the spatial maps and histograms of programmed weights and corresponding errors (defined as the difference between W_{EXP} and W_{TGT}), presented in Fig. 4a-c for the hidden layer (16×16) and in Fig. 4d-f for the output layer (8×16): the error RMS values remain lower than $750 \mu\text{V}/\mu\text{s}$ for the hidden layer and than $980 \mu\text{V}/\mu\text{s}$ for the output layer.

However, as the temperature changes, the experimental weights change drastically, since 1T-FG cells operate in sub-threshold, where the current varies exponentially with the

temperature due to the linear decrease of threshold voltage V_{th} with temperature of about $-1 \text{ mV}^{\circ}\text{C}$ [48]. The scatter plot of Fig. 4g highlights this dependence, showing how the weight error can exceed 75% as T rises by only 30°C above T_0 . The impact of temperature can be attributed to three main effects: weight magnification, distribution bending, and spreading.

We exploit an adaptive variation with temperature of the voltage amplitude V_{READ} to compensate for such degradation: according to Fig. 4h, obtained for $T = 30^{\circ}\text{C}$, the effect of the voltage amplitude variation on the weights is similar but opposite with respect to the temperature. This is essentially due to sub-threshold biasing of the 1T-FG transistors, where the current is proportional to $\exp[(V_{GS-READ} - V_{th})/(mk_B T)]$, where $V_{GS-READ}$ is the partition of V_{READ} applied between gate and source through capacitive coupling, k_B is Boltmann's constant and m is the MOSFET body effect coefficient.

The benefits of an adaptive voltage amplitude are well visible taking Fig. 4i as a starting point, which shows the W_{EXP} strong temperature dependence at constant V_{READ} , for both hidden and output layers. We were able to quantify weight magnification and ENOB degradation at varying temperature and V_{READ} in Fig. 4j and k, respectively: a temperature-insensitive unitary magnification factor, as well as a partial weight-ENOB compensation, can be obtained if the V_{READ} is linearly scaled with temperature, with a $\Delta V_{READ}/\Delta T$ correction of $-3 \text{ mV}^{\circ}\text{C}$, as indicated by the red lines in Fig. 4j,k. After correction, the new weight curves are reported in Fig. 4l, showing a clear and beneficial impact on the weight resilience to temperature.

The impact of temperature of weight transformation is composed of linear and non-linear (weight-dependent) factors. For a given weight value, the almost linear magnification observed in Fig. 4i (and quantified in Fig. 4j) represents the dominant, first-order effect. In Fig. 4g, as second order effects, a slight nonlinear distribution bending is observed, as the weight at a temperature different from T_0 gets distorted through a nonlinear factor ($W_{[T]} \propto (W_{[T_0]})^{T_0/T}$). In addition, we have observed a weight spread in temperature: if we program two cells to the same $W_{[T_0]}$ value, their $W_{[T]}$ values can differ due to a mismatch in the temperature response (this is a non-deterministic behavior, thus it cannot be compensated as it is unpredictable). To quantify linear magnification on one side, and non-linear magnification and random spread on the other side, we have extracted the magnification factor and weight ENOB in Fig. 4j and k, respectively. Note that the weight ENOB is calculated after performing a linear fit of the experimental weights to the target values, in order to isolate only second-order effects. The observed weight ENOB degradation can be thus ascribed to just second-order effects. Thus, although the slight curvature bending is not visually prominent in the scatter plots in Fig. 4g, it is quantitatively captured by the ENOB plot in Fig. 4k.

Our compensation technique is thus effective in tackling both first and second order effects: it maintains an optimized weight scaling across a broad temperature range (Fig. 4j) while partially correcting the distortions introduced by second-order effects (Fig. 4k). Further details on these aspects are described in Appendix C.

IV. EXPERIMENTAL DEMONSTRATION OF TEMPERATURE-RESILIENT CLASSIFICATION ACCURACY

The low-resolution (4×4) MNIST test set presented in Fig. 1, with digits from ‘1’ to ‘8’, has been used as a testbench for the characterization of the full neural network chip. The trained ANN parameters have been mapped into two TD-VMM cores within our CMOS chip, by programming the hidden-layer and output-layer weights (Fig. 4a and d, respectively) at $T = 30^\circ\text{C}$ and $V_{\text{READ}} = 1.15$ V. The

programming operations were performed by targeting a weight ENOB of 4.5 bits (see Fig. 4k) in the nominal conditions.

Two main inference test analyses have been performed, to verify the impact of temperature- and time-dependent degradation of the accuracy performance. As for the temperature behavior, we have tested the chip inference accuracy in the $10 \div 60^\circ\text{C}$ range, by comparing two approaches: on one side, we have varied the temperature by keeping all signal scaling factors and V_{READ} constant, to emphasize the impact of the

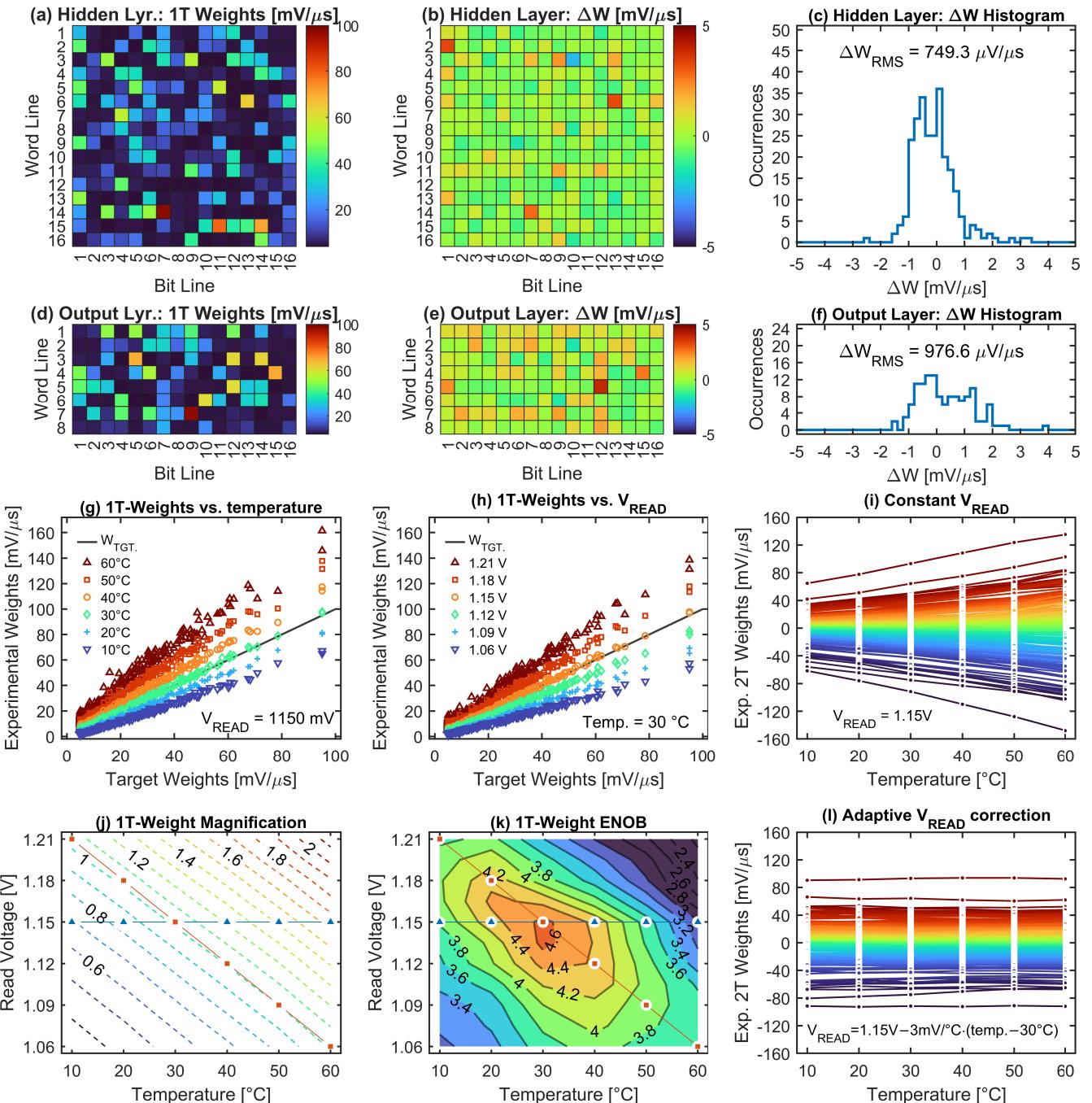


Fig. 4. Programmed Weights. Hidden Layer: spatial maps of the (a) experimental 1T-weights and (b) corresponding $\Delta W = W_{\text{EXP}} - W_{\text{TGT}}$ and (c) related error histogram. Output Layer: spatial maps of the (d) experimental 1T-weights and (e) corresponding ΔW and (f) related error histogram. Scatter plots of $1T - W_{\text{EXP}}$ (hidden+output layers) with respect to $1T - W_{\text{TGT}}$ as a function of (g) temperature and of (h) V_{READ} . (j) Weight magnification and (k) ENOB as a function of temperature and V_{READ} . 2T-weight dependence of the temperature under (i) constant V_{READ} and (l) V_{READ} correction conditions.

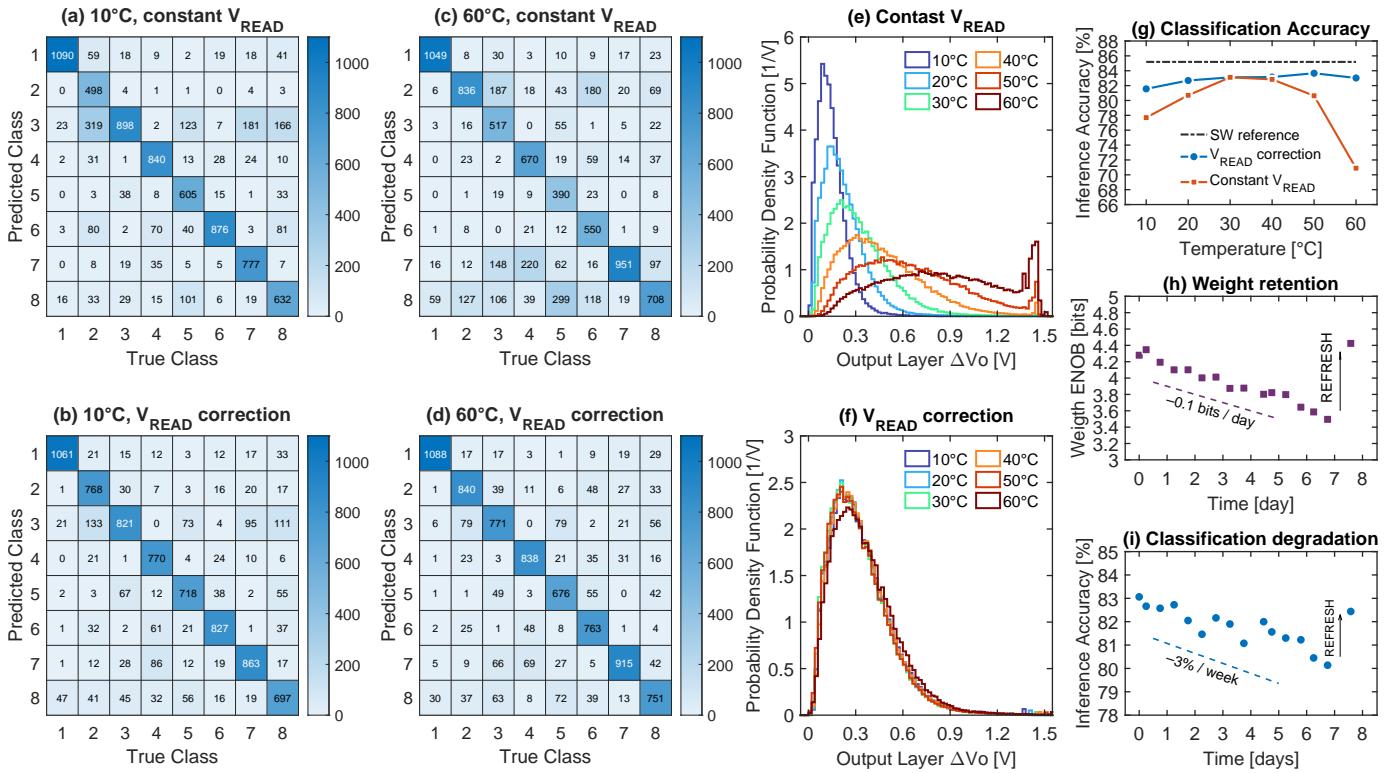


Fig. 5. Low-resolution digit classification. Confusion matrices extracted under constant V_{READ} and adaptive V_{READ} tests at 10°C ((a) and (b), respectively) and 60°C ((c) and (d), respectively). Corresponding normalized histograms (in probability distribution function form) of the output-layer TD-VMM outputs under (e) constant V_{READ} and (f) V_{READ} correction tests. (g) Classification accuracy against temperature. Degradation over time of the (h) weight ENOB and of the (i) network classification accuracy.

natural temperature-sensitivity of the 1T-FG memory cells storing the weights; on the other side, the same analysis has been repeated by using the adaptive V_{READ} with the linear correction rule ($\Delta V_{\text{READ}}/\Delta T = -3\text{mV}/\text{°C}$).

As a preliminary remark, one should note that, in principle, a linear scaling of the weights would not affect the classification performance in the implemented neural network architecture. However, when the weights magnification is not accompanied by a corresponding rescaling of the input data, the integrator circuits are pushed towards saturation, as limited by the supply voltage. In addition, a further non-linear effect is introduced by the fact that weight transformation with temperature is, in fact, not exactly linear (see again Fig. 3k).

Fig. 5a-d report the confusion matrices of the individual handwritten digit tests at 10°C (a,b) and 60°C (c,d), comparing the two cases of constant and corrected V_{READ} , where we see a clear reduction of inference errors when using an adaptive voltage amplitude. In the constant V_{READ} mode, at 10°C (Fig. 5a), the degradation is mainly attributed to images reporting the digit ‘2’, which are often erroneously classified as a ‘3’ by the network (in 319 cases, representing a -4% of inference accuracy loss). Instead, at 60°C (Fig. 5c), the mistakes are distributed within all the digits (e.g. 299 ‘5’ are classified as ‘8’, 220 ‘4’ are classified as ‘7’, and so on). Note that the occurrences of all these typical mistakes are reduced by a good amount when the V_{READ} correction method is implemented (Fig. 5b and d).

For all the investigated temperatures, Fig. 5e reports the

histograms of the output-layer VMM outputs in the fixed- V_{READ} condition, making it clear how the temperature causes a dramatic deformation of the expected results if no countermeasures are taken: low temperature brings the output distribution towards low voltages (blue histogram), thus lowering the signal-to-noise ratio; on the other hand, the upper tail of the distribution at high temperature falls in the saturation operation of the charge amplifiers (brown histogram), thus introducing a high degree of distortion. On the contrary, when the $V_{\text{READ}}(T)$ correction is exploited, the histogram of the outputs taken at all the investigated temperatures are well overlapped, as evidenced by Fig. 5f.

The measured inference accuracy, as obtained by testing 8000 images of the MNIST test-data low-resolution version, is reported in Fig. 5g, and reaches a typical value of 83% at nominal conditions (versus an 85% for a double-precision floating-point software implementation). Without the adaptive $V_{\text{READ}}(T)$, the classification accuracy of our chip falls down to 77.7% at 10°C and to 70.9% at 60°C. However, the adaptive $V_{\text{READ}}(T)$ results in a temperature-resilient accuracy, with all values in the 30°C–60°C range slightly higher than 83% and a minimum of 81.56% measured at 10°C.

Finally, the classification accuracy performance of the programmed chip has been measured over time for a week at 20°C, to investigate the retention of the memory cores storing the analog weights. Fig. 5h report the ENOB extracted considering both hidden and output weights: an ENOB degradation of approximately -0.1 bits/day has been measured, being the

cause of the $-3\%/\text{week}$ classification accuracy degradation shown in Fig. 5i. The physical nature of the FG memory core allows us to make a memory refresh, which can be exploited to bring the weights back close to the target, leading to a recovered classification capability of the neuromorphic chip. A silicon process with a thicker oxide (e.g. 10 nm) would provide industrial-grade retention.

V. CONCLUSION

Our results show that it is possible to use proper circuit and device design techniques to overcome some of the main weaknesses of analog computing circuits, i.e. sensitivity to non linearity, noise and temperature variations, while still preserving the main strengths of analog computing, i.e. low-power operation and the intrinsic massive parallelism, while providing sufficient precision for achieving high classification accuracy in analog neural network classifiers. Therefore, we believe these results can pave the way towards the exploitation of low-cost CMOS foundry processes to design and manufacture analog neuromorphic chips. In addition, this demonstration and the reliability and accessibility of consolidated CMOS processes are a very good starting point to achieve very-low-energy per inference in large artificial intelligence models implemented with analog neuromorphic chips.

APPENDIX

A. Analog NVM 1T-FG cell programming and read

Fig. 1c illustrates the programming operation for a single 1T-FG cell. A write pulse adds negative charge to the floating polysilicon gate, in order to reduce the cell conductivity and therefore the stored weight: this is obtained by applying a drain voltage V_{wri} between 5.5 V and 6.5 V, causing channel hot electron injection (CHEI) into the polysilicon gate, where electrons are trapped (Fig. 1c, left). An erase pulse adds positive charge to the floating polysilicon gate, to neutralize the negative charge and to increase cell conductivity (increase the stored weight): a drain voltage V_{ers} higher than 7 V causes impact-ionizing hot hole injection (IHII) into the floating gate (Fig. 1c, right). In all cases, the source is at ground [14].

In order to program the analog weights, we adopt a program and verify scheme, cyclically applying short write/erase pulses and extracting the weights for verification, and then repeating the cycle until the desired current is obtained. The weight cells are prone to programming half-selection disturbance due to the crossbar memory architecture, which does not include selectors to isolate unaddressed cells. While this issue is somehow tolerable in relatively small arrays, and can be corrected with a successive-refinement program and verify iterative algorithm, it is likely to become critical in larger crossbar array implementations. In such cases, we believe that an isolation mechanism, such as a selector for each weight cell, would be necessary to prevent cross-coupling during programming oper-

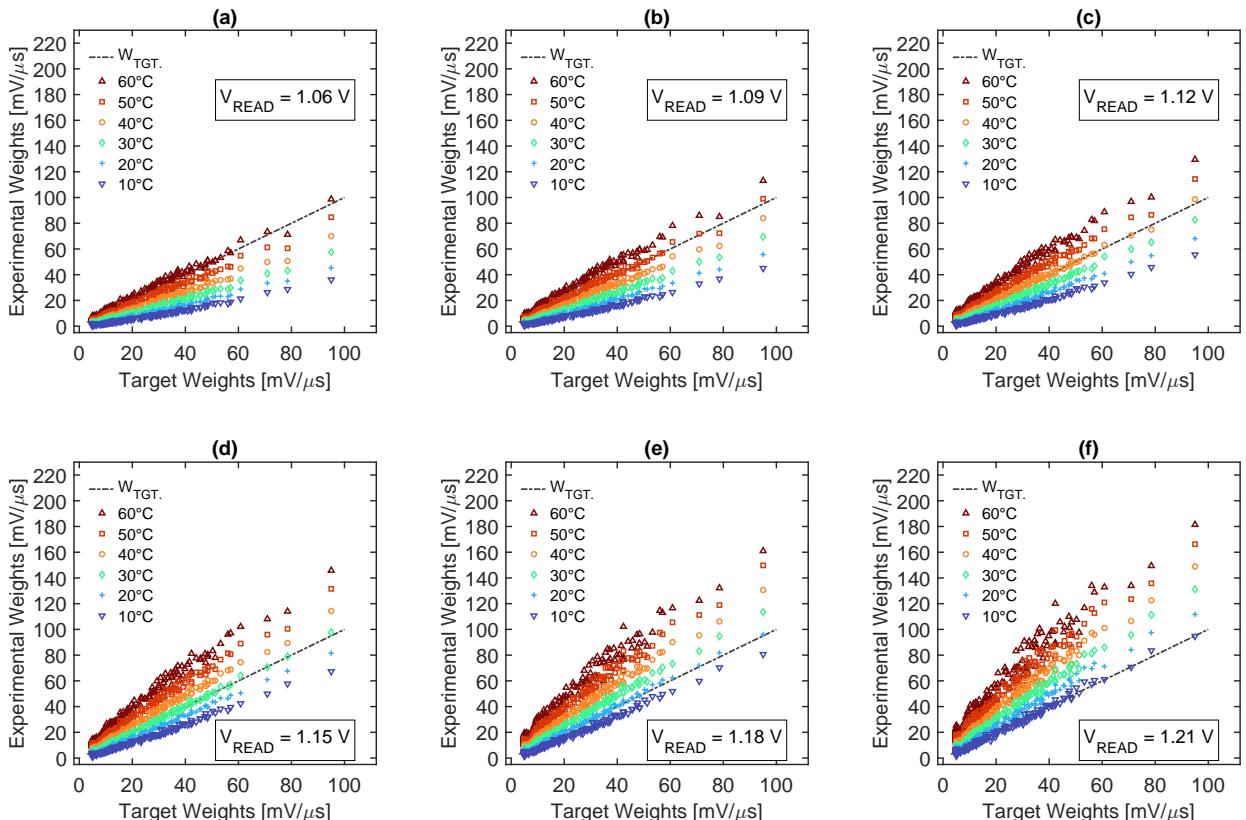


Fig. 6. Hidden Layer: weight scatter plots for different V_{READ} and temperature conditions. (a) 1.06 V, (b) 1.09 V, (c) 1.12 V, (d) 1.15 V, (e) 1.18 V, (f) 1.21 V (temperature from 10°C to 60°C, with 10°C step).

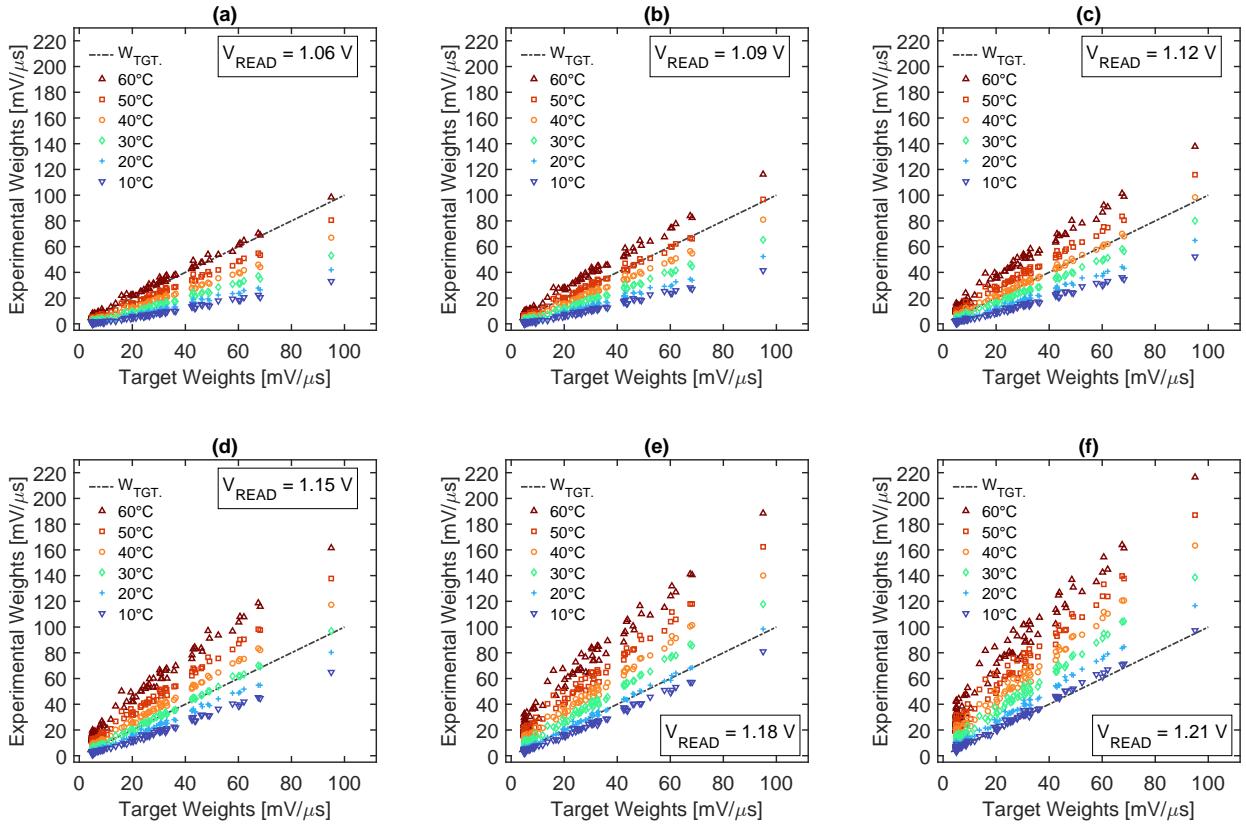


Fig. 7. **Output Layer:** weight scatter plots for different V_{READ} and temperature conditions. (a) 1.06 V, (b) 1.09 V, (c) 1.12 V, (d) 1.15 V, (e) 1.18 V, (f) 1.21 V (temperature from 10°C to 60°C, with 10°C step).

ations. In this case, we have exploited a partial countermeasure during weight extraction, which is carried out this way: i) a set of random pulse-width input vectors (PW_{IN}) are sent to the chip to perform the VMM operations; ii) based on the measured results (analog $\Delta V_{\text{OUT},\text{EXP}}$), the weight matrix W is extracted by finding the argument W_{EXP} which minimizes the function $RMS(\Delta V_{\text{OUT},\text{EXP}} - PW_{IN} \times W_{\text{EXP}})$. Interestingly, since these PW_{IN} input vectors are drawn from a distribution with a shape similar to that used during the actual inference test (i.e. we used images from the training dataset), this approach helps to compensate for some of the systematic offsets and non-idealities. It can thus be regarded as a form of hardware-aware weight transfer.

B. Time Domain Vector-Matrix Multiplier Operation

The TD-VMM cores can operate in three different modes: “VMM” for normal VMM compute-in-memory operation; “PROGRAM” and “READ” for memory multi-level programming and reading. For a closer look at the “VMM” mode, we can refer to Fig. 2a, showing the details of a VMM column implementation. In this operation mode, the decoders are deactivated, since all the 16×16 matrix cells operate in parallel; the drivers make sure that each row – word-line (WL), connected to the 1T-FG cell source terminals – sees its relative input time pulse, which activates with a fixed drain-to-source voltage the corresponding row of 1T-FG cells for the time of its duration. Details on the timing of the VMM operation are given in Fig. 2b, hereafter further discussed.

“VMM” mode - Input timing: The 16 input pulse-width signals are generated by an array of digital-to-pulse converters. The input data are loaded in 16 8-bit registers (with signal $LOAD$ active), and are then transferred to 16 8-bit counters. The MSB of each register and counter is always pre-loaded with ‘1’, while the other 7 bits (which we name $D_{IN,i}$, with ‘ i ’ from 1 to 16) contain the desired i -th pulse duration value in terms of clock periods, $T_{\text{CLK}} = 250$ ns (minimum limited by the evaluation board microcontroller, as a standard GPIO pin is used to provide the clock to the analog chip). When the $LOAD$ signal is deactivated and the VMM operation starts (low-to-high transition of the gov_{VMM} signal, at time $t_{0,\text{VMM}}$) the counters start counting and each corresponding MSB_i goes low after $(128 - D_{IN,i}) \times T_{\text{CLK}}$. The 16 MSB_i lines are buffered and used to implement the 16 pulse-width WLs, as they stay low for a time equal to $D_{IN,i} \times T_{\text{CLK}}$, being reset to ‘1’ all together at the end of 128 $T_{\text{CLK}s}$ after $t_{0,\text{VMM}}$.

“VMM” mode - Output timing: The corresponding output voltage waveform of a column integrator is also shown. The integrator is reset at the beginning of each VMM operation (with res signal), and its output voltage has a piece-wise-linear waveform with the slope increasing at the beginning of each WL pulse. The $V_{\text{OUT},j}$ of each column is sampled before the integration (after reset) – $V_{\text{RST},j}$ – and at the end of the operation – $V_{\text{END},j}$ –, and the j -th output result of the VMM is given by the voltage difference $\Delta V_{\text{OUT},j} = V_{\text{END},j} - V_{\text{RST},j}$.

“PROGRAM” mode: Fig. 1c displays the 1T-FG single cell architecture highlighting the physical phenomena which

enable program and erase operations. In “PROGRAM” mode, the BL is forced to ground, whereas a programming pulse is provided to the WL (with amplitude V_{prog}). Non-selected WLs and BLs are kept in high impedance to limit half-selection issues. The programming voltage pulse V_{prog} can activate different injection phenomena depending on its intensity, thus writing or erasing the cell. For a write operation, a V_{wri} voltage as high as 5.5 V to 6.5 V and typical pulse width of 20 ms causes the tunneling of hot electrons from the channel to the gate, where they remain trapped, leading to an increase of the threshold voltage V_{th} of the cell. For even higher V_{prog} values, erasing occurs, since the accelerated electrons generate electron-hole pairs in the channel and the high V_{DS} favors hole injection through the gate oxide into the floating gate, resulting in a V_{th} reduction. To get an effective erase we have used 9 V V_{ers} pulses of 100 μs . 1T-FG cell program is a demanding operation from both time and energy point of view. Total programming time, $T_{\text{Full-Prog}}$: tens of iterations \times number of weights $\times t_{\text{prog}}$, with typical t_{prog} of 100 μs (reset) up to 20 ms (set); total programming energy, $E_{\text{Full-Prog}}$: $T_{\text{Full-Prog}} \times V_{\text{prog}} \times I_{\text{prog}}$, with typical V_{prog} and I_{prog} of 6 V (set) to 9 V (reset) and ~ 1 mA, respectively.

“READ” mode: the “READ” mode enables the physical access to the terminals of each single cell, as the selected WL and BL are connected to the external circuitry via the read_{WL} and read_{BL} lines. The programmed state of the selected cell can be read by sourcing a voltage to the read_{WL} and sensing the current at the read_{BL} while forcing it at ground, or vice versa.

TD-VMM analog compute accuracy in terms of ENOB: In order to experimentally verify the VMM accuracy performance, we have measured the native 16×16 weights of our TD-VMM cores (weight extraction method detailed in Appendix - A), with a typical chip weight map reported in Fig. 2c.

After the extraction, a set of 512×16 pulse-width input vectors (inset of Fig. 2c) has been sent to the chip. The analog core, by processing the given inputs, has executed the 512×16 TD-VMM operations, and the histogram of the resulting ΔV_{OUT} curves is compared in Fig. 2d with the one constructed from theoretically expected results. The same experimental and theoretical data are used to obtain the scatter plot in Fig. 2e, where each color refers to a different VMM BL column. The histogram of Fig. 2f is built considering the errors related to all the 16 BL outputs, from which we were able to extract an overall RMS error (defined as $\Delta V_{OUT,\text{EXP}} - \Delta V_{OUT,\text{REF}}$) of 10.21 mV. Thus, by considering the RMS $\Delta V_{OUT,\text{REF}}$ value of 648.2 mV, we extract an aggregate ENOB of the VMM of 5.7 bits [ENOB = $(\text{SNDR}_{\text{dB}} - 1.76)/6.02$], where SNDR is here calculated as the ratio of the $\Delta V_{OUT,\text{REF}}$ RMS value to the RMS error. The same approach has been used in Fig. 2g to compute the ENOB for each independent BL column of the TD-VMM, from three separate dies, with all of them showing a typical value of the ENOB close to 5.7 bits.

C. Temperature dependence of experimentally programmed weights

In this section we analyze in detail the temperature impact on programmed weights. Due to the sub-threshold operation of 1T-FG cells, experimental weights strongly depend on the operating temperature. We try to explain this relation with three main effects, which are: (1) a weight magnification when varying the temperature; (2) a bending of the weight distribution curvature (which specifically depends on the target weight); (3) and distribution spread, which does not depend on the target weight. Ideally, a pure weight (de)magnification applied to the whole network layer parameters would not have direct impact on the weight ENOB, and would not impact the classification accuracy of our classifier. However, in the hardware implementation, for a fixed set of input data, the weight magnification with increasing temperature moves the corresponding outputs in a voltage region where the precision of the VMM is degraded, for instance towards the charge amplifier saturation. This issue could be compensated by rescaling the input data, as long as the pulse-width rescaling does not push the TD-VMMs beyond their bandwidth limits. On the other hand, both bending and spread have a direct impact on the weight ENOB, as the actual temperature-related magnification of each weight depends on its corresponding nominal weight value (resulting in distortion), and weights programmed to the same target value at a nominal temperature, spread out when T is changed. Adapting a dynamic read voltage can compensate for both magnification and distortion, exploiting the fact that the cells are biased in sub-threshold region. Indeed, the read voltage operates on the weights in the opposite direction with respect to the temperature, i.e. if V_{READ} is reduced, we can observe both a weight compression and a curvature of the scatter plot, that can contrast the magnification and bending induced by the temperature raise. In Fig. 6 and Fig. 7 we report the 1T-FG experimentally programmed unsigned weights for the hidden- and output-layer, respectively. Scatter plots of experimental weights plotted against the respective theoretical target are reported for different V_{READ} and temperature conditions. From these figures, we are able to generate the plots for weight magnification and ENOB of the hidden+output layers unsigned weights of Fig. 4j and k, respectively. The corresponding $2T-W_{\text{EXP}}$ dependence on the temperature, when a constant V_{READ} is used, is shown in Fig. 4i. After exploiting a $V_{\text{READ}} = 1.15 \text{ V} - 3 \text{ mV}/^{\circ}\text{C} \cdot (T - 30^{\circ}\text{C})$ correction rule, such dependence is strongly reduced, as reported in Fig. 4l.

D. Low resolution MNIST test: TD-VMM outputs

The low-resolution (4×4) MNIST test set presented in Fig. 1, with digits from ‘1’ to ‘8’, has been executed to characterize the full neural network chip, as summarized in Fig. 5. To this purpose, data reported in Fig. 5e-f (normalized histograms of the output-layer TD-VMM outputs as obtained for varying temperature and at constant or varying V_{READ}), have been extrapolated from Fig. 8 and Fig. 9. In these figures, we report the outputs of the TD-VMM cores when processing the low resolution MNIST test at the hidden- and output-layer, respectively. The scatter plots have been

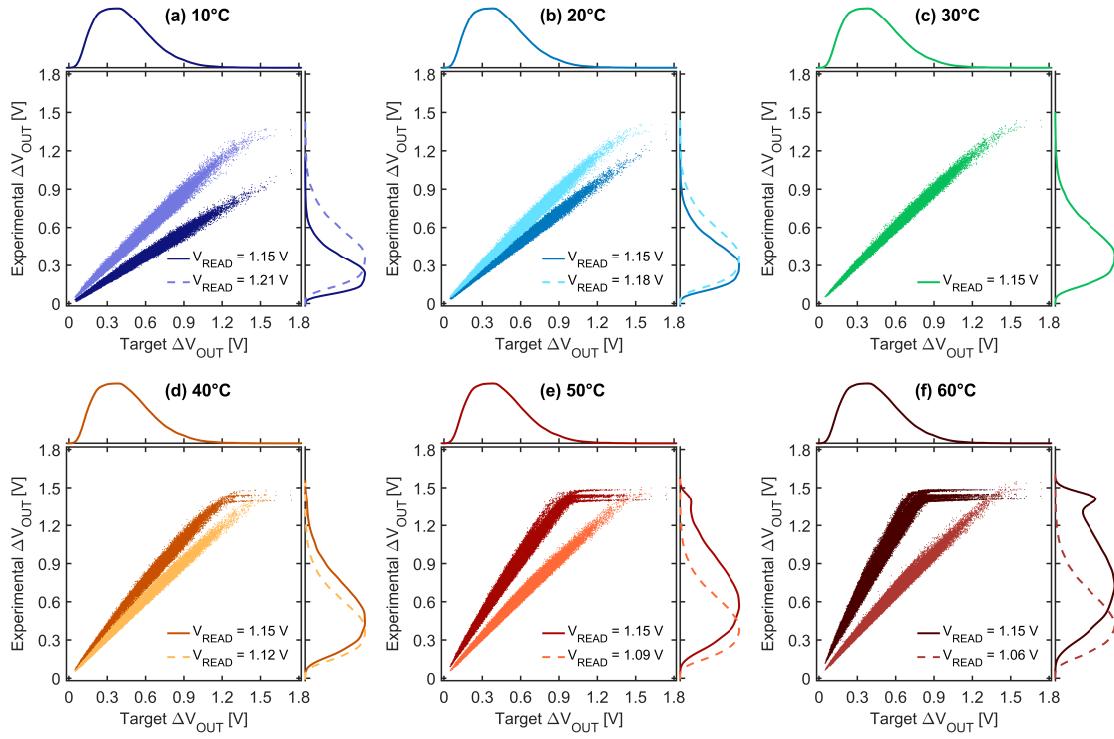


Fig. 8. Experimental hidden-layer VMM output scatter plots and distributions. Scatter plot of the experimental ΔV_{OUT} plotted against the respective target (obtained through software simulation) for temperature from 10°C (a) to 60°C (f). Two options are reported for each temperature: constant $V_{READ} = 1.15$ V case, showing the drift of the distributions towards low levels of the output range (low temperature) or towards the output range saturation (high temperature); V_{READ} correction case ($\Delta V_{READ}/\Delta T = -3 \text{ mV/}^{\circ}\text{C}$) showing the obtained resilience of the output distributions against the temperature variations. Reported data points have been computed by the hardware VMM by processing 8000 low-resolution MNIST test-images.

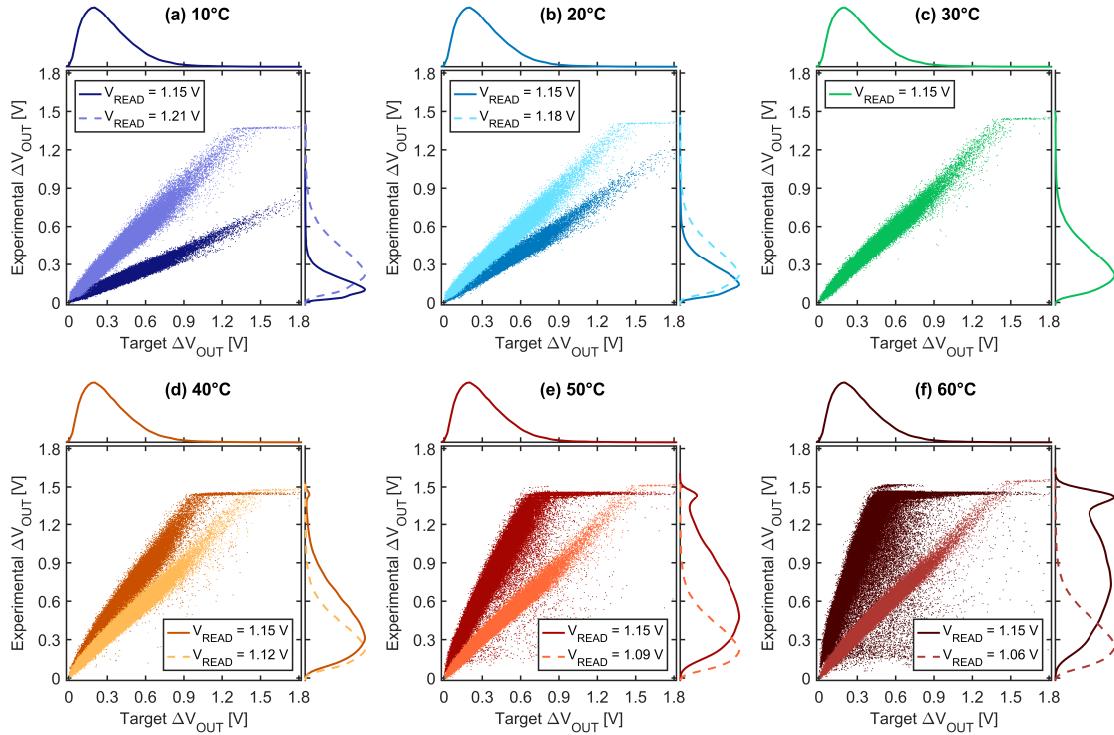


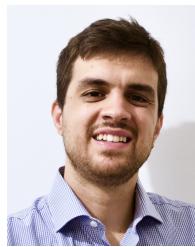
Fig. 9. Experimental output-layer VMM output scatter plots and distributions. Scatter plot of the experimental ΔV_{OUT} plotted against the respective target (obtained through software simulation) for temperatures from 10°C (a) to 60°C (f). Two options are reported for each temperature: constant $V_{READ} = 1.15$ V case, showing the drift of the distributions towards low levels of the output range (low temperature) or towards the output range saturation (high temperature); V_{READ} correction case ($\Delta V_{READ}/\Delta T = -3 \text{ mV/}^{\circ}\text{C}$) showing the obtained resilience of the output distributions against the temperature variations. Reported data points have been computed by the hardware VMM by processing 8000 low-resolution MNIST test-images.

constructed by plotting the VMM outputs obtained for all 8000 test images. The test on the full dataset has been repeated for 6 temperatures, from 10°C up to 60°C (10°C step), by considering a constant V_{READ} read mode (dark data points) and a V_{READ} linear temperature correction method (light data point).

REFERENCES

- [1] X. Xu, Y. Ding, S. X. Hu, M. Niemier, J. Cong, Y. Hu, and Y. Shi, “Scaling for edge inference of deep neural networks,” *Nature Electronics*, vol. 1, pp. 216–222, 4 2018.
- [2] J. Bian, Z. Cao, and P. Zhou, “Neuromorphic computing: Devices, hardware, and system application facilitated by two-dimensional materials,” *Applied Physics Review*, vol. 8, no. 4, p. 041313, 12 2021.
- [3] K. Berggren, Q. Xia, K. K. Likharev, D. B. Strukov, H. Jiang, T. Mikolajick, D. Querlioz, M. Salinga, J. R. Erickson, S. Pi, F. Xiong, P. Lin, C. Li, Y. Chen, S. Xiong, B. D. Hoskins, M. W. Daniels, A. Madhavan, J. A. Liddle, J. J. McClelland, Y. Yang, J. Rupp, S. S. Nonnenmann, K.-T. Cheng, N. Gong, M. A. Lastras-Montaño, A. A. Talin, A. Salleo, B. J. Shastri, T. F. de Lima, P. Prucnal, A. N. Tait, Y. Shen, H. Meng, C. Roques-Carmes, Z. Cheng, H. Bhaskaran, D. Jariwala, H. Wang, J. M. Shainline, K. Segall, J. J. Yang, K. Roy, S. Datta, and A. Raychowdhury, “Roadmap on emerging hardware and technology for machine learning,” *Nanotechnology*, vol. 32, p. 012002, 1 2021.
- [4] D. V. Christensen, R. Dittmann, B. Linares-Barranco, A. Sebastian, M. Le Gallo, A. Redaelli, S. Slesazeck, T. Mikolajick, S. Spiga, S. Menzel, I. Valov, G. Milano, C. Ricciardi, S.-J. Liang, F. Miao, M. Lanza, T. J. Quill, S. T. Keene, A. Salleo, J. Grollier, D. Marković, A. Mizrahi, P. Yao, J. J. Yang, G. Indiveri, J. P. Strachan, S. Datta, E. Vianello, A. Valentian, J. Feldmann, X. Li, W. H. P. Pernice, H. Bhaskaran, S. Furber, E. Neftci, F. Scherr, W. Maass, S. Ramaswamy, J. Tapson, P. Panda, Y. Kim, G. Tanaka, S. Thorpe, C. Bartolozzi, T. A. Cleland, C. Posch, S. Liu, G. Panuccio, M. Mahmud, A. N. Mazumder, M. Hosseini, T. Mohsenin, E. Donati, S. Tolu, R. Galeazzi, M. E. Christensen, S. Holm, D. Ielmini, and N. Pryds, “2022 roadmap on neuromorphic computing and engineering,” *Neuromorphic Comput. Eng.*, vol. 2, no. 2, p. 022501, 6 2022.
- [5] W. Zhang, P. Yao, B. Gao, Q. Liu, D. Wu, Q. Zhang, Y. Li, Q. Qin, J. Li, Z. Zhu, Y. Cai, D. Wu, J. Tang, H. Qian, Y. Wang, and H. Wu, “Edge learning using a fully integrated neuro-inspired memristor chip,” *Science*, vol. 381, no. 6663, pp. 1205–1211, 2023.
- [6] W. Zhang, B. Gao, J. Tang, P. Yao, S. Yu, M.-F. Chang, H.-J. Yoo, H. Qian, and H. Wu, “Neuro-inspired computing chips,” *Nature Electronics*, vol. 3, no. 7, pp. 371–382, Jul 2020.
- [7] D. Marković, A. Mizrahi, D. Querlioz, and J. Grollier, “Physics for neuromorphic computing,” *Nature Reviews Physics*, vol. 2, no. 9, pp. 499–510, Sep 2020.
- [8] Q. Wei, B. Gao, J. Tang, H. Qian, and H. Wu, “Emerging memory-based chip development for neuromorphic computing: Status, challenges, and perspectives,” *IEEE Electron Devices Magazine*, vol. 1, no. 2, pp. 33–49, 2023.
- [9] Z. Sun, S. Kvatin斯基, X. Si, A. Mehonic, Y. Cai, and R. Huang, “A full spectrum of computing-in-memory technologies,” *Nature Electronics*, vol. 6, no. 11, pp. 823–835, Nov 2023.
- [10] F. Merrikh-Bayat, X. Guo, M. Klachko, M. Prezioso, K. K. Likharev, and D. B. Strukov, “High-Performance Mixed-Signal Neurocomputing With Nanoscale Floating-Gate Memory Cell Arrays,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4782–4790, 2018.
- [11] L. Danial, E. Pikhay, E. Herbelin, N. Wainstein, V. Gupta, N. Wald, Y. Roizin, R. Daniel, and S. Kvatin斯基, “Two-terminal floating-gate transistors with a low-power memristive operation mode for analogue neuromorphic computing,” *Nature Electronics*, vol. 2, no. 12, pp. 596–605, Dec 2019.
- [12] M. Paliy, S. Strangio, P. Ruiu, T. Rizzo, and G. Iannaccone, “Analog Vector-Matrix Multiplier Based on Programmable Current Mirrors for Neural Network Integrated Circuits,” *IEEE Access*, vol. 8, pp. 203 525–203 537, 2020.
- [13] W. Wang, L. Danial, Y. Li, E. Herbelin, E. Pikhay, Y. Roizin, B. Hoffer, Z. Wang, and S. Kvatin斯基, “A memristive deep belief neural network based on silicon synapses,” *Nature Electronics*, vol. 5, no. 12, p. 870 – 880, 2022.
- [14] T. Rizzo, S. Strangio, and G. Iannaccone, “Time Domain Analog Neuromorphic Engine Based on High-Density Non-Volatile Memory in Single-Poly CMOS,” *IEEE Access*, vol. 10, pp. 49 154–49 166, 2022.
- [15] W. Haensch, T. Gokmen, and R. Puri, “The next generation of deep learning hardware: Analog computing,” *Proceedings of the IEEE*, vol. 107, no. 1, pp. 108–122, 2019.
- [16] A. Mehonic and A. J. Kenyon, “Brain-inspired computing needs a master plan,” *Nature*, vol. 604, no. 7705, pp. 255–260, Apr 2022.
- [17] C. D. Schuman, S. R. Kulkarni, M. Parsa, J. P. Mitchell, P. Date, and B. Kay, “Opportunities for neuromorphic computing algorithms and applications,” *Nature Computational Science*, vol. 2, no. 1, pp. 10–19, Jan 2022.
- [18] N. J. Tye, S. Hofmann, and P. Stanley-Marbell, “Materials and devices as solutions to computational problems in machine learning,” *Nature Electronics*, vol. 6, no. 7, pp. 479–490, Jul 2023.
- [19] P. Mannocci, M. Farronato, N. Lepri, L. Cattaneo, A. Glukhov, Z. Sun, and D. Ielmini, “In-memory computing with emerging memory devices: Status and outlook,” *APL Machine Learning*, vol. 1, no. 1, p. 010902, 02 2023.
- [20] A. D. Patil, H. Hua, S. Gonugondla, M. Kang, and N. R. Shanbhag, “An mram-based deep in-memory architecture for deep neural networks,” in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2019, pp. 1–5.
- [21] P. Rzeszut, J. Chęciński, I. Brzozowski, S. Ziętek, W. Skowroński, and T. Stobiecki, “Multi-state mram cells for hardware neuromorphic computing,” *Scientific Reports*, vol. 12, no. 1, p. 7178, May 2022.
- [22] M.-K. Kim, I.-J. Kim, and J.-S. Lee, “Cmos-compatible compute-in-memory accelerators based on integrated ferroelectric synaptic arrays for convolution neural networks,” *Science Advances*, vol. 8, no. 14, p. eabm8537, 2022.
- [23] E. Coví, H. Mulaosmanovic, B. Max, S. Slesazeck, and T. Mikolajick, “Ferroelectric-based synapses and neurons for neuromorphic computing,” *Neuromorphic Computing and Engineering*, vol. 2, no. 1, p. 012002, feb 2022.
- [24] H.-H. Lin, C.-C. Lin, C.-T. Shih, W.-Y. Jang, and T.-Y. Tseng, “Mgo/hzo based ferroelectric tunnel junctions for neuromorphic computing applications,” *IEEE Electron Device Letters*, vol. 44, no. 9, pp. 1444–1447, 2023.
- [25] S. Thomann, A. Mema, K. Ni, and H. Amrouch, “Reliable fetef-based neuromorphic computing through joint modeling of cycle-to-cycle variability, device-to-device variability, and domain stochasticity,” in *2023 IEEE International Reliability Physics Symposium (IRPS)*, 2023, pp. 1–5.
- [26] G. Migliato Marega, Z. Wang, M. Paliy, G. Giusi, S. Strangio, F. Castiglione, C. Callegari, M. Tripathi, A. Radenovic, G. Iannaccone, and A. Kis, “Low-Power Artificial Neural Network Perceptron Based on Monolayer MoS2,” *ACS Nano*, vol. 16, no. 3, pp. 3684–3694, Mar 2022.
- [27] G. Migliato Marega, H. G. Ji, Z. Wang, G. Pasquale, M. Tripathi, A. Radenovic, and A. Kis, “A large-scale integrated vector–matrix multiplication processor based on monolayer molybdenum disulfide memories,” *Nature Electronics*, Nov 2023.
- [28] C. Du, F. Cai, M. A. Zidan, W. Ma, S. H. Lee, and W. D. Lu, “Reservoir computing using dynamic memristors for temporal information processing,” *Nature Communications*, vol. 8, no. 1, p. 2204, Dec 2017.
- [29] M. Farronato, P. Mannocci, M. Melegari, S. Ricci, C. M. Compagnoni, and D. Ielmini, “Reservoir Computing with Charge-Trap Memory Based on a MoS2 Channel for Neuromorphic Engineering,” *Advanced Materials*, vol. 35, no. 37, 2023.
- [30] Q. Huo, Y. Yang, Y. Wang, D. Lei, X. Fu, Q. Ren, X. Xu, Q. Luo, G. Xing, C. Chen, X. Si, H. Wu, Y. Yuan, Q. Li, X. Li, X. Wang, M.-F. Chang, F. Zhang, and M. Liu, “A computing-in-memory macro based on three-dimensional resistive random-access memory,” *Nature Electronics*, vol. 5, no. 7, pp. 469–477, Jul 2022.
- [31] W. Wan, R. Kubendran, C. Schaefer, S. B. Eryilmaz, W. Zhang, D. Wu, S. Deiss, P. Raina, H. Qian, B. Gao, S. Joshi, H. Wu, H.-S. P. Wong, and G. Cauwenberghs, “A compute-in-memory chip based on resistive random-access memory,” *Nature*, vol. 608, no. 77923, pp. 504–512, Aug 2022.
- [32] Z. Sun, G. Pedretti, A. Bricalli, and D. Ielmini, “One-step regression and classification with cross-point resistive memory arrays,” *Science Advances*, vol. 6, no. 5, p. eaay2378, 2020.
- [33] S. Wang, Y. Luo, P. Zuo, L. Pan, Y. Li, and Z. Sun, “In-memory analog solution of compressed sensing recovery in one step,” *Science Advances*, vol. 9, no. 50, p. eadj2908, 2023.
- [34] R. Khaddam-Aljameh, M. Stanisavljevic, J. Fornt Mas, G. Karunaratne, M. Braendli, F. Liu, A. Singh, S. Muller, U. Egger, A. Petropoulos, T. Antonakopoulos, K. Brew, S. Choi, I. Ok, F. Lie, N. Saulnier, V. Chan,

- I. Ahsan, V. Narayanan, S. Nandakumar, M. Le Gallo, P. Francese, A. Sebastian, and E. Eleftheriou, "HERMES Core-A 14nm CMOS and PCM-based In-Memory Compute Core using an array of 300ps/LSB Linearized CCO-based ADCs and local digital processing," in *IEEE Symposium on VLSI Circuits, Digest of Technical Papers*, 2021.
- [35] M. Le Gallo, R. Khaddam-Aljameh, M. Stanisavljevic, A. Vasilopoulos, B. Kersting, M. Dazzi, G. Karunaratne, M. Brändli, A. Singh, S. M. Müller, J. Büchel, X. Timoneda, V. Joshi, M. J. Rasch, U. Egger, A. Garofalo, A. Petropoulos, T. Antonakopoulos, K. Brew, S. Choi, I. Ok, T. Philip, V. Chan, C. Silvestre, I. Ahsan, N. Saulnier, V. Narayanan, P. A. Francese, E. Eleftheriou, and A. Sebastian, "A 64-core mixed-signal in-memory compute chip based on phase-change memory for deep neural network inference," *Nature Electronics*, vol. 6, no. 9, p. 680 – 693, 2023.
- [36] M. Rao, H. Tang, J. Wu, W. Song, M. Zhang, W. Yin, Y. Zhuo, F. Kiani, B. Chen, X. Jiang, H. Liu, H.-Y. Chen, R. Midya, F. Ye, H. Jiang, Z. Wang, M. Wu, M. Hu, H. Wang, Q. Xia, N. Ge, J. Li, and J. J. Yang, "Thousands of conductance levels in memristors integrated on CMOS," *Nature*, vol. 615, no. 7954, p. 823 – 829, 2023.
- [37] E. P. Yakov Roizin, "Memristor using parallel asymmetrical transistors having shared floating gate and diode." 2016, u.s. Patent 9,514,818.
- [38] D. Fick, "Analog compute-in-memory for ai edge inference," in *IEEE International Electron Device Meeting, Digest of Technical Papers*. Institute of Electrical and Electronics Engineers (IEEE), 1 2022, pp. 21.8.1–21.8.4.
- [39] C. Huang and S. Chakrabarty, "A temperature compensated array of cmos floating-gate analog memory," in *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, 2010, pp. 109–112.
- [40] X. Guo, F. M. Bayat, M. Prezioso, Y. Chen, B. Nguyen, N. Do, and D. B. Strukov, "Temperature-insensitive analog vector-by-matrix multiplier based on 55 nm nor flash memory cells," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, 2017, pp. 1–4.
- [41] A. Dilello, S. Andryzcik, B. M. Kelly, B. Rumberg, and D. W. Graham, "Temperature compensation of floating-gate transistors in field-programmable analog arrays," in *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2017, pp. 1–4.
- [42] D. Wei, J. Qu, Y. Zeng, and L. Qiao, "A compensation principle for the temperature sensitivity of in-memory computing arrays," in *2024 13th Non-Volatile Memory Systems and Applications Symposium (NVMSA)*, 2024, pp. 1–6.
- [43] D. C. Monga, O. Numan, M. Andraud, and K. Halonen, "A temperature and process compensation circuit for resistive-based in-memory computing arrays," in *2023 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2023, pp. 1–5.
- [44] J.-O. Seo, M. Seok, and S. Cho, "A 44.2-tops/w cnn processor with variation-tolerant analog datapath and variation compensating circuit," *IEEE Journal of Solid-State Circuits*, vol. 59, no. 5, pp. 1603–1611, 2024.
- [45] G. Singh, O. Numan, D. Monga, M. Andraud, and K. Halonen, "On-chip built-in self-calibration of thermal variations for mixed-signal in-memory computing," in *2024 IEEE European Test Symposium (ETS)*, 2024, pp. 1–6.
- [46] "The MNIST Database of handwritten digits," URL: <http://yann.lecun.com/exdb/mnist/>, [Online; accessed 2023-09-10].
- [47] W. Kester, "Analog Devices MT-003 Tutorial: Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR," URL: <https://www.analog.com/media/en/training-seminars/tutorials/mt-003.pdf>, [Online; accessed 2025-03-18].
- [48] J. Tzou, C. Yao, R. Cheung, and H. Chan, "The temperature dependence of threshold voltages in submicrometer cmos," *IEEE Electron Device Letters*, vol. 6, pp. 250–252, 5 1985.



Tommaso Rizzo received the M.S. degree (2019), and the Ph.D. degree (2023) in EE from the University of Pisa, both cum laude. He obtained his Ph.D. in a joint program with Quantavis s.r.l., Pisa, working as a research engineer, with a thesis on analog IC design for implantable neuromorphic devices. From 2014 to 2019, he was "Allievo Ordinario" at Sant'Anna School of Advanced Studies, Pisa. In 2017, he was with Fermilab, Batavia, IL, USA, and in 2019, he joined imec, Eindhoven, NL, both as a visiting student. He is currently with STMicroelectronics, Pisa, working in the advanced analog R&D product division. His research focuses on advanced analog IC design using standard and non-standard CMOS technologies.



Sebastiano Strangio received the B.S. and M.S. degrees (cum laude) in EE, and the Ph.D. degree from the University of Calabria, Cosenza, Italy, in 2010, 2012, and 2016, respectively. He was with IMEC, Leuven, Belgium, as a Visiting Student, in 2012, working on the electrical characterization of resistive-RAM memory cells. From 2013 to 2016, he was a Temporary Research Associate with the University of Udine, and with the Forschungszentrum Jülich, Germany, as a Visiting Researcher, in 2015, researching on TCAD simulations, design, and characterization of TFET-based circuits. From 2016 to 2019, he was with LFoundry, Avezzano, Italy, where he worked as a Research and Development Process Integration and Device/TCAD Engineer, with main focus on the development of a CMOS Image Sensor Technology Platform. He is currently a Researcher in electronics with the University of Pisa. He has authored or coauthored over 40 papers, most of them published in IEEE journals and conference proceedings. His research interests include technologies for innovative devices (e.g. TFETs) and circuits for innovative applications (CMOS analog building blocks for DNNs), as well as CMOS image sensors, power devices and circuits based on wide-bandgap materials.



Alessandro Catania received the B.S., M.S., and Ph.D. degrees in electronic engineering from the University of Pisa, Italy, in 2014, 2016 and 2020, respectively. He is currently working as an Assistant Professor with the University of Pisa. His current research interests include mixed-signal microelectronic design for harsh environments and wireless power transfer systems for implantable systems.



Giuseppe Iannaccone received the M.S. and Ph.D. degrees in EE from the University of Pisa, in 1992 and 1996, respectively. He is currently Deputy President and Professor of electronics with the University of Pisa. He has coordinated several European and national projects involving multiple partners and has acted as principal investigator in several research projects funded by public agencies at the European and national level, and by private organizations. He co-founded the academic spinoff Quantavis s.r.l. and is involved in other technology transfer initiatives. He has authored or coauthored more than 250 articles published in peer-reviewed journals and more than 160 papers in proceedings of international conferences, gathering more than 12000 citations on the Scopus database. His research interests include quantum transport and noise in nanoelectronic and mesoscopic devices, development of device modeling tools, new device concepts and circuits beyond CMOS technology for artificial intelligence, cybersecurity, implantable biomedical sensors, and the Internet of Things. He is a fellow of the American Physical Society.